ALLEVIATING TIMING BASED CONGESTION WITHIN CIRCUIT DESIGNS

BACKGROUND

Field of the Invention

[0001] This invention relates to the field of integrated circuits and, more particularly, to accommodating timing-based congestion within such devices.

Description of the Related Art

[0002] Very Large Scale Integrated (VLSI) circuits such as application specific integrated circuits (ASIC's) and field programmable gate arrays (FPGA's), have become increasingly complex and heterogeneous. Modern integrated circuits can include a variety of different components or resources including, but not limited to, block random access memories (RAM's), multipliers, processors, logic blocks, and the like. These components must be interconnected using a variety of different routing resources. The routing resources, however, are finite. As such, competition among components for wiring resources can dictate many aspects of an integrated circuit design such as area usage and timing performance.

[0003] Competition for wiring resources can be referred to as congestion. Modeling resource-based congestion has become an important aspect of integrated circuit design. A resource-based congestion model attempts to depict congestion within the integrated circuit design without regard for timing characteristics. Several resource-based congestion models have been proposed in the context of integrated circuit design. These models have included empirical models such as Rent's Rule, stochastic models that utilize probabilistic approaches, as well as the detection of congestion using a global router.

[0004] With respect to integrated circuit design, however, timing constraints and timing characteristics must be taken into account. In illustration, for an integrated circuit to function properly, particular connections and paths must

EV000337478US

conform to predetermined timing constraints. Additionally, some resources such as wires have more favorable timing characteristics than others. Such is the case as wires available within an integrated circuit can be of varying length and can be made of different materials. Thus, a signal propagating through one wire can require more or less time depending upon the length of the wire and the type of material from which the wire was made. There tends to be increased competition among components for wires having more favorable signal propagation characteristics, particularly in light of timing constraints.

[0005] Whenever demand for timing efficient resources exceeds the supply, timing-based congestion results. What is needed is a technique for dealing with timing-based congestion in the context of integrated circuit design.

SUMMARY OF THE INVENTION

[0006] The present invention provides a method, system, and apparatus for detecting and alleviating timing-based congestion. Based upon an initial routing of an integrated circuit design, an initial delay associated with each connection of the circuit design can be determined. A final delay for the connections can be predicted based upon the congestion, or competition for wiring resources within the circuit design. Accordingly, the final delay prediction can be evaluated with respect to any timing constraints of the circuit design to determine whether to perform a detailed routing of the circuit design or optimize the circuit design further.

[0007] One embodiment of the present invention can include a method of relieving timing-based congestion during physical implementation of an integrated circuit. The method can include routing a placed circuit design of the integrated circuit in a delay mode. An initial delay for connections of the circuit design can be calculated based upon the routing step. A final delay for connections of the circuit design can be predicted were connection overlaps to be removed. Connections of the circuit design that do not conform with timing constraints based upon at least one of the initial and final delays can be identified. A detailed routing of the circuit design or further optimization of the identified connections of the circuit design can be performed based upon the identifying step.

[0008] Other embodiments of the present invention can include a machine readable storage for causing a machine to perform the steps disclosed herein as well as a system for performing the various steps disclosed herein.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] There are shown in the drawings, embodiments which are presently preferred, it being understood, however, that the invention is not limited to the precise arrangements and instrumentalities shown.

[0010] FIG. 1 is a schematic diagram illustrating one embodiment of a system for relieving timing-based congestion of a circuit design.

[0011] FIG. 2 is a flow chart illustrating a method of relieving timing-based congestion of a circuit design in accordance with another embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0012] FIG. 1 is a schematic diagram illustrating one embodiment of a system 100 for relieving timing-based congestion of a circuit design for an integrated circuit. In accordance with the inventive arrangements disclosed herein, system 100 can be used to place and route integrated circuits such as FPGA's and ASIC's. As shown, the system 100 can include a placer 110, a quick router 115, a timing engine 120, a congestion detection module 125, and a detailed router 130.

[0013] The placer 110 can load a circuit design 105A, such as a netlist or other circuit representation specifying

components, functionality, and connectivity. The placer 110 performs an initial assignment of components to physical locations on the circuit. The initially placed circuit design can be provided to the quick router 115.
The quick router 115 receives the circuit design and identifies one or more vectors of signals to be routed. Accordingly, the quick router 115 performs an initial routing of signals in the circuit design. Further details on the quick router are disclosed in U.S. Patent Application Serial No. 10/637,242, entitled "Using Routing Feedback for Placement Improvements for Logic Design," by Sankaranarayanan Srinivasan, et. al., filed August 7, 2003, which is herein incorporated by reference.

[0014] Signals between a source and load, referred to as a connection, typically can be routed in one of two modes, a resource mode or a delay mode. In resource mode, connections are routed between source and load using a minimum number of wires or resources. Resource mode does not minimize the propagation delay for a connection.

[0015] By comparison, in delay mode, connections are routed such that the signal propagation delay is minimized. Unless the placement of a circuit design is altered, a connection routed in delay mode cannot be optimized any further in terms of signal propagation delay. The delay mode, however, permits overlap conditions to occur. An overlap condition refers to the case where more than one signal shares a single wire, an infeasible condition.

[0016] The quick router 115 performs the initial routing using the delay mode which allows overlap conditions to occur. While overlap conditions lead to an infeasible routing, valuable timing information can be determined from such a state. In one embodiment of the present invention the quick router routes the timing critical connections in the delay mode and the non-critical or non-timing sensitive connections in a non-delay mode, e.g., resource mode.

[0017] The circuit design is then passed to the timing engine 120. The timing engine 120 performs a timing analysis upon the initial routing of the circuit design. The timing analysis yields timing information which can include, but is not limited to, slack values for connections as well as connection and/or path delays. Slack values indicate the degree by which a connection meets, exceeds, or fails a target delay for a connection as specified by the timing constraints. Thus, for example, a zero value indicates the timing constraint for a connection is met exactly, a negative slack indicates a failing connection, and a positive slack indicates a passing connection. The more negative the slack, the greater the degree by which the connection fails the target delay. The higher the value of a positive slack, the greater the degree by which the connection exceeds the target delay.

[0018] The timing engine 120 further can compare the timing information for each connection with the target delay associated with that connection as specified by the predetermined timing constraints. Accordingly, the timing engine 120 can determine whether the circuit design meets the timing constraints.

[0019] The congestion detection module 125 can calculate predictions of connection delays assuming overlap conditions have been removed. The connections delays predicted by the congestion detection module 125 can be provided back to the timing engine 120, which then can perform a slack/criticality calculation based upon the predicted connection delay values.

[0020] Notably, with respect to evaluating delays for connections, connections can be categorized into one of three different classes. The first class of connection is where the initial delay of the connection fails its timing requirement. In that case, since the connection has already been routed in delay mode, the predicted delay will also fail the timing requirement because the predicted delay cannot be less than the already optimized delay of the connection.

Connections of this variety will fail timing requirements even if no congestion/overlap exists with respect to resources used by the connections. These connections can only be alleviated through placement changes or physical synthesis.

[0021] The second class of connection includes connections where both the initial timing delay and the predicted final delay of the connection pass the timing constraint. In other words, a prediction has been made that such connections will meet associated timing constraints after a detailed routing is performed. Notably, if all connections of the design are categorized as belonging to the second class of connections, the circuit design can be successfully routed in a manner than meets all timing constraints without any need for congestion alleviation.

[0022] The third class includes connections where the initial delay passes the timing constraint, but the predicted final delay of the connection fails the timing constraint. This third class of connection is representative of the effects of timing-based congestion. The connections corresponding to this scenario must be addressed.

[0023] If the timing analysis reveals that one or more connections are not meeting the timing constraints the circuit design can be further optimized by the placer 110 and the quick router 115. More particularly, the placer 110 can perform incremental placement optimizations on the circuit design with respect to components associated with failing connections. An incremental or minor change to a circuit design can be one in which an insubstantial number of components is changed or moved. For example, in one embodiment, an incremental change can be one that alters less than approximately 10% of the components of a circuit design. Still, it should be appreciated that incremental placement changes can include re-packing lookup tables (LUT's) and flip-flops as well as moving components to new locations.

[0024] Thus, the placer 110 can identify any connections which do not meet the timing constraints as determined by the timing engine 130. Components associated with failing connections also can be identified. The placer 110 can relocate one or more components in an effort to reduce the delay associated with failing connections with the goal of meeting or exceeding the timing constraints.

[0025] With respect to components sharing a failing connection belonging to the first class, timing constraints can be met by moving the components closer to one another. With respect to components sharing a failing connection belonging to the third class, such components can be relocated to another area of the circuit design where less competition for routing resources exists. It should be appreciated, however, that indiscriminately moving components closer to one another can lead to increased connection failures for the third class of connections, as moving components closer together can produce increased competition for limited routing resources.

[0026] The circuit design, having one or more components assigned to new positions, can be provided to the quick router 115. The quick router 115 can then reroute only those connections that have been affected by the relocation of components by the placer 110.

[0027] While the system 100 can operate on the placement of the circuit design in an iterative fashion, once a determination is made by the timing engine 120 that the connections meet or exceed the timing constraints, the placed circuit design can be provided to the detailed router 130. The detailed router 130 then can perform a detailed or final routing to produce circuit design 105B. More particularly, the detailed router 130 can operate on the circuit design to route signals in accordance with predetermined or programmed design constraints and timing information determined during the placement phase. In one embodiment of the present invention, the detailed router 130, rather than begin perform an entirely new routing, can receive the overlapped solution as input and operate on the overlapped solution to produce a non-overlapping, or legal, solution.

[0028] In any event, the detailed router 130 can route signals, verify that the routing complies with established design constraints, and ensure that the resulting routing is feasible. In other words, the detailed router 130 does not utilize an overlap mode.

[0029] The placer 110, the quick router 115, the timing engine 120, and the detailed router 130 can be implemented as application programs executing within a suitable computer and/or information processing system. It should be appreciated that while each is depicted in FIG. 1 as a separate entity, one or more of the application programs can be combined into a larger, more complex application. For example, the quick router 115 and the placer 110 can be combined if so desired. In any case, those skilled in the art will recognize that the examples presented herein are not intended as a limitation of the present invention.

[0030] FIG. 2 is a flow chart illustrating a method 200 of relieving timing-based congestion of a circuit design in accordance with another embodiment of the present invention. The method 200 can begin in step 205 where a circuit design for an integrated circuit is loaded into a circuit design system as described herein. In step 210, the circuit design can be placed. In step 215, a quick routing of the circuit design is performed using a delay mode.

[0031] At this point, one or more competing connections may be using a single wire of the circuit design. If so, the routing is an infeasible one. To produce a feasible routing, each wire within the circuit design must have only a single connection using that wire. Such a condition can be referred to as overlap free. Still, the overlap condition provides an indication as to the timing-based congestion of the circuit design.

[0032] In step 220, the timing-based congestion of the circuit design can be determined based upon the quick routing. More particularly, the number of contenders, i.e., connections competing, or assigned, to a wire can be determined for one or more, or all, wires of the circuit design. Thus, for each given wire, the number of contending connections for that wire can be obtained. In step 225, the initial delay for each connection of the circuit design can be determined. A connection is an abstract concept connecting two modules in a circuit design and includes one or more wires, that are the actual implementation of the connection on the integrated circuit such as the programmable interconnection resources of an FPGA. Hence each connection is routed using one or more wires and a wire on the integrated circuit can be shared by multiple connections in the overlap mode.

[0033] In step 230, a prediction of the final delays for connections can be determined. The phrase "final delay" for a connection refers to the delay of the connection after overlap conditions have been removed. This final delay value for one or more, or all, of the connections of the circuit design can be predicted without having to actually reroute the circuit design without overlaps. Accordingly, a designer can, based upon the final delay predictions, determine whether a circuit design is likely to meet or fail timing constraints.

[0034] The final delays for connections can be determined by first calculating the number of contenders for a connection. One way to determine the number of contenders for a connection (i.e., *NumContendersForConnection*) is to sum the number of connections sharing each wire (i.e., *NumContenderForWire*) of a given connection. In another embodiment, a weighted sum can be determined where the weighting factor is derived by analyzing the delay for a particular wire or by analyzing the relative timing criticalities of the connections using the wire. In any case, the number of contenders for a connection can be determined using the following formula:

$NumContendersForConnection = \sum \alpha_i (NumContenderForWire)$, where

α_i can be the weighting factor for the ith wire. Notably, the weighting factor α_i can be set equal to 1, for all i, in the case where no weighting factor is desired.

[0035] In yet another embodiment, the weighting factor can be determined by analyzing the importance of a wire during the overlap removal phase. A history parameter can be defined to record the congestion on a wire, i.e. the number of connections using the wire, during overlap removal iterations. In the overlap removal phase, the connections on a wire can be removed based upon the criticality of the connection and re-routed by increasing the cost of the wires in the device.

[0036] Accordingly, the final delay of a connection can be calculated using the following formula:

$FinalDelay = InitialDelay(1+\beta * NumContendersForConnection)$,

where β is a factor which, along with the number of contenders for a connection, determines the degree to which the connection will degrade in terms of delay due to congestion. In one embodiment, the factor β can be determined by performing a statistical analysis on the initial and final delay values for a set of timing constrained connections in different circuit designs.

[0037] In another embodiment, this factor can be dynamically generated during design implementation by executing one or more iterations of an overlap removal technique or routine in the router where some nets are ripped up and rerouted. As such, those nets undergo a degradation in delay. Still, in yet another embodiment, the factor β can be implemented as a time changing function.

[0038] It should be appreciated, however, that other models may be used to predict the final delay of a connection. For example, a quadratic model can be used. As such, the present invention is not limited to the use of one particular type of model, as the linear model presented herein has been used for purposes of illustration only.

[0039] In step 235, a timing analysis can be performed using the predicted delays for the circuit design connections. In step 240, a determination can be made as to whether predetermined timing constraints regarding the circuit design, and specifically for each connection, have been met. If so, the method can proceed to step 245 to perform a detailed routing of the circuit design. Once complete, the method can end.

[0040] If, however, the timing constraints for one or more connections have not been met, the connections of the circuit design that do not conform with timing constraints based upon at least one of the initial and final delays can be identified. Accordingly, the method can proceed to step 250 where incremental placement of the circuit design can be performed. In one embodiment, only those components associated with connections that do not conform with timing constraints need be operated upon. In step 255, incremental quick routing can be performed. The method can loop back to step 220 to determine the timing-based congestion of the circuit design and repeat as necessary.

[0041] It should be appreciated that while the method can end when all connections comply with the timing constraints, other termination criteria can be used as well. For example, in one embodiment, the method 200 can terminate if a solution is not found after a predetermined number of iterations. In another embodiment, the method 200 can terminate if the timing characteristics do not improve by a predetermined percentage from one iteration to the next. For example, improvement can be measured by the number of failing connections from one iteration to the next.

[0042] The present invention provides a method, system, and apparatus for detecting timing-based congestion. In accordance with the inventive arrangements disclosed herein, after a quick routing of the circuit design is performed, an initial delay for one or more connections of the circuit design can be determined. A final delay for the connections can be predicted based upon the initial delay and an analysis of the timing-based congestion of the circuit design. The final delays are used to perform a timing analysis of the circuit design. Based upon whether the circuit design meets imposed timing constraints, the circuit design can be optimized or provided to a detailed router.

[0043] The present invention can be realized in hardware, software, or a combination of hardware and software. The present invention can be realized in a centralized fashion in one computer system, or in a distributed fashion where different elements are spread across several interconnected computer systems. Any kind of computer system or other apparatus adapted for carrying out the methods described herein is suited. A typical combination of hardware and software can be a general purpose computer system with a computer program that, when being loaded and executed, controls the computer system such that it carries out the methods described herein.

[0044] The present invention also can be embedded in a computer program product, which comprises all the features enabling the implementation of the methods described herein, and which when loaded in a computer system is able to carry out these methods. Computer program in the present context means any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after either or both of the following: a) conversion to another language, code or notation; b) reproduction in a different material form.

[0045] This invention can be embodied in other forms without departing from the spirit or essential attributes thereof. Accordingly, reference should be made to the following claims, rather than to the foregoing specification, as indicating the scope of the invention.